

# Exam

## Section 1

### Fill in the Blanks

1. The \_\_\_\_\_ Design Pattern is a creational design pattern that provides an interface for creating families of related or dependent objects without specifying their concrete classes.
2. \_\_\_\_\_ interface declares a set of methods to create abstract product objects.
3. \_\_\_\_\_ Factories implement the abstract factory interface, producing families of related products.
4. \_\_\_\_\_ Products declare interfaces for a set of distinct products, each produced by a family of concrete factories.
5. The client uses the abstract factory and abstract product interfaces to create families of \_\_\_\_\_ objects.

## Section 2

### Identification

1. What design pattern provides an interface for creating families of related or dependent objects without specifying their concrete classes?
2. What is the common theme that individual factories encapsulate in the Abstract Factory Pattern?
3. What does the Abstract Factory interface declare?
4. What do Concrete Factories implement in the Abstract Factory Pattern?
5. What do Abstract Products declare in the Abstract Factory Pattern?

6. Who implements the interfaces defined by abstract products in the Abstract Factory Pattern?
7. What is ideal for managing dependencies between multiple interrelated objects in the Abstract Factory Pattern?
8. What is useful in systems that need to be configured with multiple families of related objects in the Abstract Factory Pattern?
9. In the Abstract Factory Pattern, when the system needs to be configured with one of the multiple variants of a product family, what is this an example of?
10. In the Abstract Factory Pattern, what does the Abstract Factory Interface declare for each type of abstract product?

### **Section 3**

#### **Essay**

1. How does the Abstract Factory Pattern ensure the consistency of created objects?
2. What are the advantages of using the Abstract Factory Pattern in terms of product creation flexibility?
3. Can you discuss the encapsulation benefits provided by the Abstract Factory Pattern?
4. What are the potential drawbacks of implementing the Abstract Factory Pattern in a system with numerous product families?
5. How does the Abstract Factory Pattern facilitate the management of dependencies between multiple interrelated objects?